WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Tuesday, August 17, 2004

Hide?	Set Name	Query	Hit Count
		DB=USPT; PLUR=YES; OP=ADJ	
☐	L15	(assign$4 or locat$4) same (ip or internet protocol) same location and (map$4 same address$2) and L8 and quer$4 and domain$ and (dns or domain name server)	39
☐	L14	(assign$4 or locat$4) same (ip or internet protocol) same location	3928
☐	L13	(map$4 same address$2) and L8 and quer$4 and domain$ and (dns or domain name server)	39
☐	L12	(map$4 same address$2) and L8 and quer$4 and domain$	39
☐	L11	(map$4 same address$2) and L8	39
☐	L10	(map$4 and addess$2) and L8	0
☐	L9	(map$4 same addess$2) and L8	0
☐	L8	(geolocation or geograph$4) and L7	40
☐	L7	map$4 and L6	44
☐	L6	host and server and L5	51
☐	L5	api and L4	51
☐	L4	(dns or domain name server) and L3	94
☐	L3	query and domain$ and L2	148
☐	L2	network and l1	437
☐	L1	assign$4 same (ip or internet protocol) same location	458

END OF SEARCH HISTORY

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 39 of 39 returned.

☐ 1. Document ID: US 6778524 B1

L15: Entry 1 of 39 File: USPT Aug 17, 2004

US-PAT-NO: 6778524
DOCUMENT-IDENTIFIER: US 6778524 B1

TITLE: Creating a geographic database for network devices

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 2. Document ID: US 6769000 B1

L15: Entry 2 of 39 File: USPT Jul 27, 2004

US-PAT-NO: 6769000
DOCUMENT-IDENTIFIER: US 6769000 B1

TITLE: Unified directory services architecture for an IP mobility architecture framework

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 3. Document ID: US 6754181 B1

L15: Entry 3 of 39 File: USPT Jun 22, 2004

US-PAT-NO: 6754181
DOCUMENT-IDENTIFIER: US 6754181 B1

TITLE: System and method for a directory service supporting a hybrid communication system architecture

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 4. Document ID: US 6742015 B1

L15: Entry 4 of 39 File: USPT May 25, 2004

US-PAT-NO: 6742015
DOCUMENT-IDENTIFIER: US 6742015 B1

TITLE: Base services patterns in a netcentric environment

Full	Title	Citation	Front	Review	Classification	Date	Reference			Claims	KWIC	Draw De

5. Document ID: US 6731625 B1

L15: Entry 5 of 39 File: USPT May 4, 2004

US-PAT-NO: 6731625
DOCUMENT-IDENTIFIER: US 6731625 B1

TITLE: System, method and article of manufacture for a call back architecture in a hybrid network with support for internet telephony

Full	Title	Citation	Front	Review	Classification	Date	Reference			Claims	KWIC	Draw De

6. Document ID: US 6715145 B1

L15: Entry 6 of 39 File: USPT Mar 30, 2004

US-PAT-NO: 6715145
DOCUMENT-IDENTIFIER: US 6715145 B1

TITLE: Processing pipeline in a base services pattern environment

Full	Title	Citation	Front	Review	Classification	Date	Reference			Claims	KWIC	Draw De

7. Document ID: US 6684250 B2

L15: Entry 7 of 39 File: USPT Jan 27, 2004

US-PAT-NO: 6684250
DOCUMENT-IDENTIFIER: US 6684250 B2

TITLE: Method and apparatus for estimating a geographic location of a networked entity

Full	Title	Citation	Front	Review	Classification	Date	Reference			Claims	KWIC	Draw De

8. Document ID: US 6640249 B1

L15: Entry 8 of 39 File: USPT Oct 28, 2003

US-PAT-NO: 6640249
DOCUMENT-IDENTIFIER: US 6640249 B1
**** See image for Certificate of Correction ****

TITLE: Presentation services patterns in a netcentric environment

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

---

☐ 9. Document ID: US 6640244 B1

L15: Entry 9 of 39 File: USPT Oct 28, 2003

US-PAT-NO: 6640244
DOCUMENT-IDENTIFIER: US 6640244 B1

TITLE: Request batcher in a transaction services patterns environment

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

---

☐ 10. Document ID: US 6640238 B1

L15: Entry 10 of 39 File: USPT Oct 28, 2003

US-PAT-NO: 6640238
DOCUMENT-IDENTIFIER: US 6640238 B1
**** See image for Certificate of Correction ****

TITLE: Activity component in a presentation services patterns environment

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

---

☐ 11. Document ID: US 6636242 B2

L15: Entry 11 of 39 File: USPT Oct 21, 2003

US-PAT-NO: 6636242
DOCUMENT-IDENTIFIER: US 6636242 B2

TITLE: View configurer in a presentation services patterns environment

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

---

☐ 12. Document ID: US 6615253 B1

L15: Entry 12 of 39 File: USPT Sep 2, 2003

US-PAT-NO: 6615253
DOCUMENT-IDENTIFIER: US 6615253 B1
**** See image for Certificate of Correction ****

TITLE: Efficient server side data retrieval for execution of client side applications

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

---

☐ 13. Document ID: US 6615199 B1

L15: Entry 13 of 39 File: USPT Sep 2, 2003

US-PAT-NO: 6615199
DOCUMENT-IDENTIFIER: US 6615199 B1
**** See image for Certificate of Correction ****

TITLE: Abstraction factory in a base services pattern environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 14. Document ID: US 6614781 B1

L15: Entry 14 of 39 File: USPT Sep 2, 2003

US-PAT-NO: 6614781
DOCUMENT-IDENTIFIER: US 6614781 B1

TITLE: Voice over data telecommunications network architecture

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 15. Document ID: US 6606660 B1

L15: Entry 15 of 39 File: USPT Aug 12, 2003

US-PAT-NO: 6606660
DOCUMENT-IDENTIFIER: US 6606660 B1

TITLE: Stream-based communication in a communication services patterns environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 16. Document ID: US 6601234 B1

L15: Entry 16 of 39 File: USPT Jul 29, 2003

US-PAT-NO: 6601234
DOCUMENT-IDENTIFIER: US 6601234 B1

TITLE: Attribute dictionary in a business logic services environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 17. Document ID: US 6601192 B1

L15: Entry 17 of 39 File: USPT Jul 29, 2003

US-PAT-NO: 6601192
DOCUMENT-IDENTIFIER: US 6601192 B1

TITLE: Assertion component in environment services patterns

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 18. Document ID: US 6578068 B1
L15: Entry 18 of 39 File: USPT Jun 10, 2003

US-PAT-NO: 6578068
DOCUMENT-IDENTIFIER: US 6578068 B1

TITLE: Load balancer in environment services patterns

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 19. Document ID: US 6571282 B1
L15: Entry 19 of 39 File: USPT May 27, 2003

US-PAT-NO: 6571282
DOCUMENT-IDENTIFIER: US 6571282 B1

TITLE: Block-based communication in a communication services patterns environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 20. Document ID: US 6550057 B1
L15: Entry 20 of 39 File: USPT Apr 15, 2003

US-PAT-NO: 6550057
DOCUMENT-IDENTIFIER: US 6550057 B1
**** See image for Certificate of Correction ****

TITLE: Piecemeal retrieval in an information services patterns environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 21. Document ID: US 6549949 B1
L15: Entry 21 of 39 File: USPT Apr 15, 2003

US-PAT-NO: 6549949
DOCUMENT-IDENTIFIER: US 6549949 B1

TITLE: Fixed format stream in a communication services patterns environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

---

☐ 22. Document ID: US 6539396 B1
L15: Entry 22 of 39 File: USPT Mar 25, 2003

US-PAT-NO: 6539396
DOCUMENT-IDENTIFIER: US 6539396 B1
**** See image for Certificate of Correction ****

TITLE: Multi-object identifier system and method for information service pattern environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

---

☐ 23. Document ID: US 6529948 B1
L15: Entry 23 of 39 File: USPT Mar 4, 2003

US-PAT-NO: 6529948
DOCUMENT-IDENTIFIER: US 6529948 B1
**** See image for Certificate of Correction ****

TITLE: Multi-object fetch component

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

---

☐ 24. Document ID: US 6529909 B1
L15: Entry 24 of 39 File: USPT Mar 4, 2003

US-PAT-NO: 6529909
DOCUMENT-IDENTIFIER: US 6529909 B1
**** See image for Certificate of Correction ****

TITLE: Method for translating an object attribute converter in an information services patterns environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

---

☐ 25. Document ID: US 6502213 B1
L15: Entry 25 of 39 File: USPT Dec 31, 2002

US-PAT-NO: 6502213
DOCUMENT-IDENTIFIER: US 6502213 B1
**** See image for Certificate of Correction ****

TITLE: System, method, and article of manufacture for a polymorphic exception handler in environment services patterns

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Drawi De

---

☐ 26. Document ID: US 6496850 B1

L15: Entry 26 of 39 File: USPT Dec 17, 2002

US-PAT-NO: 6496850
DOCUMENT-IDENTIFIER: US 6496850 B1

TITLE: Clean-up of orphaned server contexts

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Drawi De

---

☐ 27. Document ID: US 6477665 B1

L15: Entry 27 of 39 File: USPT Nov 5, 2002

US-PAT-NO: 6477665
DOCUMENT-IDENTIFIER: US 6477665 B1

TITLE: System, method, and article of manufacture for environment services patterns in a netcentic environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Drawi De

---

☐ 28. Document ID: US 6477580 B1

L15: Entry 28 of 39 File: USPT Nov 5, 2002

US-PAT-NO: 6477580
DOCUMENT-IDENTIFIER: US 6477580 B1

TITLE: Self-described stream in a communication services patterns environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Drawi De

---

☐ 29. Document ID: US 6442748 B1

L15: Entry 29 of 39 File: USPT Aug 27, 2002

US-PAT-NO: 6442748
DOCUMENT-IDENTIFIER: US 6442748 B1

TITLE: System, method and article of manufacture for a persistent state and persistent object separator in an information services patterns environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Drawi De

☐ 30. Document ID: US 6438594 B1

L15: Entry 30 of 39 File: USPT Aug 20, 2002

US-PAT-NO: 6438594
DOCUMENT-IDENTIFIER: US 6438594 B1

TITLE: Delivering service to a client via a locally addressable interface

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 31. Document ID: US 6434628 B1

L15: Entry 31 of 39 File: USPT Aug 13, 2002

US-PAT-NO: 6434628
DOCUMENT-IDENTIFIER: US 6434628 B1

TITLE: Common interface for handling exception interface name with additional prefix and suffix for handling exceptions in environment services patterns

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 32. Document ID: US 6434568 B1

L15: Entry 32 of 39 File: USPT Aug 13, 2002

US-PAT-NO: 6434568
DOCUMENT-IDENTIFIER: US 6434568 B1

TITLE: Information services patterns in a netcentric environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 33. Document ID: US 6339832 B1

L15: Entry 33 of 39 File: USPT Jan 15, 2002

US-PAT-NO: 6339832
DOCUMENT-IDENTIFIER: US 6339832 B1

TITLE: Exception response table in environment services patterns

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 34. Document ID: US 6335927 B1

L15: Entry 34 of 39 File: USPT Jan 1, 2002

US-PAT-NO: 6335927
DOCUMENT-IDENTIFIER: US 6335927 B1

TITLE: System and method for providing requested quality of service in a hybrid network

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

---

☐ 35. Document ID: US 6332163 B1

L15: Entry 35 of 39 File: USPT Dec 18, 2001

US-PAT-NO: 6332163
DOCUMENT-IDENTIFIER: US 6332163 B1

TITLE: Method for providing communication services over a computer network system

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

---

☐ 36. Document ID: US 6289382 B1

L15: Entry 36 of 39 File: USPT Sep 11, 2001

US-PAT-NO: 6289382
DOCUMENT-IDENTIFIER: US 6289382 B1

TITLE: System, method and article of manufacture for a globally addressable interface in a communication services patterns environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

---

☐ 37. Document ID: US 5999525 A

L15: Entry 37 of 39 File: USPT Dec 7, 1999

US-PAT-NO: 5999525
DOCUMENT-IDENTIFIER: US 5999525 A

TITLE: Method for video telephony over a hybrid network

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

---

☐ 38. Document ID: US 5867495 A

L15: Entry 38 of 39 File: USPT Feb 2, 1999

US-PAT-NO: 5867495
DOCUMENT-IDENTIFIER: US 5867495 A

TITLE: System, method and article of manufacture for communications utilizing

calling, plans in a hybrid network

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

---

☐ 39. Document ID: US 5867494 A

L15: Entry 39 of 39 File: USPT Feb 2, 1999

US-PAT-NO: 5867494
DOCUMENT-IDENTIFIER: US 5867494 A

TITLE: System, method and article of manufacture with integrated video conferencing billing in a communication system architecture

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

---

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Term	Documents
IP	45477
IPS	3830
INTERNET	50672
INTERNETS	473
PROTOCOL	114573
PROTOCOLS	74019
LOCATION	815501
LOCATIONS	437745
DNS	2265
DN	11641
DOMAIN	85236
((ASSIGN$4 OR LOCAT$4) SAME (IP OR INTERNET PROTOCOL) SAME LOCATION AND (MAP$4 SAME ADDRESS$2) AND L8 AND QUER$4 AND DOMAIN$ AND (DNS OR DOMAIN NAME SERVER)).USPT.	39

There are more results than shown above. Click here to view the entire set.

Display Format: TI | Change Format

Previous Page Next Page Go to Doc#

L11: Entry 1 of 39 File: USPT Aug 17, 2004

DOCUMENT-IDENTIFIER: US 6778524 B1
TITLE: Creating a geographic database for network devices

Abstract Text (1):
A database is populated with geographic locations for network devices by providing a node on a network and making a connection into a network service provider (NSP) point of presence (POP) to obtain a connection to the network via the NSP. A message is then transmitted to the node over the network connection obtained from the NSP. The message is received at the node and a source network identifier is extracted from the message. The source network identifier is then associated with a known geographic location for the POP in a database. The foregoing steps are then repeated for multiple different POPs. Also, a database is populated with geographic locations for network devices by providing a node on a network and making a connection into a network service provider (NSP) point of presence (POP) to obtain a connection to the network via the NSP. A message is then transmitted to the node over the network connection obtained from the NSP. The message is received at the node and a source network identifier is extracted from the message. The route over the network between the node and the POP is then probed to obtain network identifiers for routers along the route. The foregoing steps are then repeated for multiple different POPs.

Brief Summary Text (3):
The present invention concerns the creation of a database that identifies geographic locations of devices connected to a network, such as devices communicating over the Internet.

Brief Summary Text (5):
The Internet is a decentralized global network of millions computers. Each computer connected to the Internet is independent and may be capable of operating as a host computer (host) that primarily provides data over the Internet or a client computer (client) that primarily receives data over the Internet. A host computer may receive a data request from any other computer on the Internet and respond to the request by transmitting any of various types of data, such as hypertext markup language (HTML) code, back to the client. A client computer may send data requests to various hosts on the Internet and then download data in response. Typically, host computers are used by information providers for various commercial, educational, or governmental purposes and are dedicated host computers (servers or Web servers).

Brief Summary Text (6):
Ordinarily, the client computers are used by individuals to connect to the Internet via an Internet Service Provider (ISP) or, more generically, a network service provider (NSP). ISPs are companies that provide access to the Internet, typically for a fee. For example, a client computer may establish a dial-in connection to an ISP over an ordinary telephone line. ISPs are also called IAPs (Internet Access Providers).

Brief Summary Text (7):
Each host and client on the Internet is identified by a unique Internet Protocol (IP) address which is a series of numbers, such as 24.130.64.154. Because the IP

address, in its numeric form, is difficult to memorize and use, a domain name may be assigned to a host and, therefore, associated with the numeric IP address. For example, a server having an address of 24.130.64.154 may be associated with domain name server.npeponis.com. It is noted that multiple IP addresses may be associated with the same domain name and, similarly, many domain names may be associated with the same IP address or addresses. A domain name server (DNS) performs the task of converting the domain names to IP addresses. Most frequently, separate domain names are not permanently assigned to individual clients but, rather, blocks of IP addresses are assigned to the ISPs that serve those clients.

Brief Summary Text (8):
FIG. 1 illustrates a client 101 communicating with a server 103. In the instant example, the client 101 first connects to its local ISP 105 (e.g., using a modem via a dial-in connection). For purposes of the current connection only, ISP 105 assigns one of its IP addresses to client 101. Upon completion of this connection, client 101 may begin communicating over the Internet. For example, the client 101 may send a request for file main.html to the server 103 having the domain name server.npeponis.com. Such a request might be initiated, for example, by the user typing http.//server.npeponis.com/main.html in the address field of a web browser running on client computer 101 and then pressing the "Enter" key. Alternatively, such a request might be initiated by the user simply clicking on a graphic, image or text item that serves as a hyperlink to that address. In response, the browser sends out one or more data packets (or datagrams) addressed to IP address 24.130.64.154 (possibly, after having obtained that IP address from a DNS), with such data packets including a request to retrieve file main.html.

Brief Summary Text (10):
Upon receipt of request 102, the server 103 typically first initiates handshaking communications to establish a TCP connection and then responds to the request by sending to the client one or more data packets that together contain the contents of the file main.html. In this manner, communications can occur between two nodes on the Internet, with TCP/IP specifying the protocols for separating each message into data packets, routing the packets between the two nodes, reassembling the packets at the destination, and verifying that each message was properly received.

Brief Summary Text (11):
Another commonly used protocol is the HyperText Transfer Protocol (HTTP) format. The HTTP format is the underlying protocol used by the World Wide Web on the Internet and defines how messages are formatted and transmitted, as well as what actions Web servers and browsers take in response to various commands.

Brief Summary Text (12):
On the Internet, most data packets, including requests and responses, need to go through several routers before they reach their final destination. Each forwarding of a packet to the next router is termed a "hop". A router (or gateway) is a device that connects one network to another. Each router includes a dynamically updated routing table that is used by the router to identify the next router to which any given packet should be forwarded. Specifically, the receiving router attempts to identify the router that is most likely to be closest (geographically and/or in terms of number of hops) to the packet's ultimate destination.

Brief Summary Text (13):
In the example of FIG. 1, client 101 sends a request 102 to server 103. The request is delivered to the server 103 via routers 105, 107, 109, 111, 113, and 115. As indicated by the ellipsis 117, the request may go through other routers as well. In other words, the request may make many hops before reaching the intended server 103. As noted above, the precise path taken by request 102 will be determined by the individual routers along the way. In the event that a receiving router determines that it is unable to forward a packet closer to its final destination, it will send a message to that effect back to the router from which it received the

packet. Then, that router will attempt to route the packet through a different router, adjust its routing table accordingly, and send a message to the router from which it received the packet. Such a situation might be temporary (e.g., in the case where a router is temporarily inoperable) or permanent (e.g., where a router has been permanently taken off line). Other communications, such as periodically broadcasting a router's entire routing table, also occur among the routers on the Internet, permitting them to coordinate their routing activities. Propagation of changes in the network topology through the various routers in the network can permit communications to occur fairly reliably, even in the presence of constantly changing network conditions. Among the tools commonly used are the Routing Information Protocol (RIP) and the Internet Control Message Protocol (ICMP).

Brief Summary Text (14):
Irrespective of the route through which the request 102 is made or the number of hops taken by the request 102, the preferred end result is the receipt of the request 102 by the host 103 and the response by server 103 sending the requested data file via the Internet. Like the request, the data file is divided among appropriately sized (e.g., using conventional algorithms to identify an appropriate size) data packets and may travel through several routers to arrive at the client 101. Generally, the route taken by the response 104 will be the same as that taken by the request 102. However, it is possible that the routing may be asymmetric, such as where a client computer transmits packets to an ISP over a conventional telephone line/modem connection but receives packets via a satellite dish, e.g., via the Direct PC network. Asymmetric routing may also occur in certain other cases, such as where a router in the link used for transmitting the request goes down before the response to the request can be transmitted; therefore, the response needs to be re-routed. In addition to asymmetric routing, it is also possible that packets traveling in a single direction (e.g., all request packets) may take different paths (multi-path routing). This may occur, for example, in the event that a router goes down while the request is being made; in addition, one or more routers in the link may be intentionally configured to route packets that are addressed to the same destination to different routers in an attempt to balance the communication load over the Internet. However, at present, both asymmetric routing and multi-path routing are considered to be unusual routing conditions.

Brief Summary Text (15):
The response to the request may contain any of a wide variety of information. However, in many instances, it would be preferable for the response to contain information that is tailored to the specific geographic region of the client 101. For example, it may be preferable for the file 104 sent as the response to the request 102 to include weather information for the geographic region in which the client 101 is located. In another example, it may be preferable for the file 104 sent as the response to the request 102 to include banner or other advertising for businesses located within driving distance of the location of the client 101. In conventional systems, the response file 104 typically may contain these types of information only when the user of the client computer 101 has already supplied information regarding his or her location to the server 103, at least once. Unfortunately, many users may not want to expend the effort necessary to type in address or even zip code information that identifies their geographic locations. Moreover, even those that are willing to do so typically will find it very inconvenient, particularly when such information may have to be supplied for each different Web site that the user visits.

Brief Summary Text (16):
The prior art has included some discussion regarding automatically gathering information concerning the geographic location of Internet clients. However, all of these techniques have certain shortcomings, most notably, relatively long delays and limited access to geographical information pertaining to nodes on the Internet.

Brief Summary Text (17):
For example, U.S. Pat. No. 5,948,061 (the "061 Patent") to Merriman et al. titled "Method of Delivery, Targeting, and Measuring Advertising Over Networks" (which is incorporated herein by reference as though set forth herein in full) notes that a trace route operation can be used in obtaining geographic information for a user. In this regard, conventional trace route operations were originally designed to troubleshoot Internet routing problems (such as routing loops) and generally function by sending out a number of probe packets, all addressed to the same target node, to identify all of the routers that forward packets between the current node and the target node. All of the probe packets are IP packets, each having a Time-To-Live (TTL) field which indicates the maximum number of hops that the IP packet can make before an ICMP Time Exceeded packet is returned.

Brief Summary Text (20):
Conventional traceroute operations can take as long as 12 seconds on average to trace an entire routing path. This additional delay can be significant, particularly when considered in connection with all of the other delays at the server and in routing messages via the Internet. Because Internet users often are impatient with slow responding Web sites, such additional delays might result in loss of visitors to a Web site.

Brief Summary Text (21):
In addition, conventional suggested techniques for geographic positioning, such as the '061 Patent, often rely on telephone directories and other available sources to obtain geographic locations for nodes on the Internet. Such sources may be incomplete and/or not as up-to-date as possible.

Brief Summary Text (23):
The present invention addresses the foregoing problems by dialing into multiple points of presence and transmitting a message a message to a fixed node on the network through each POP.

Brief Summary Text (24):
Thus, in one aspect the invention is directed to populating a database with geographic locations for network devices. A node is provided on a network and a connection is made into a network service provider (NSP) point of presence (POP) to obtain a connection to the network via the NSP. A message is then transmitted to the node over the network connection obtained from the NSP. The message is received at the node and a source network identifier is extracted from the message. The source network identifier is then associated with a known geographic location for the POP in a database. The foregoing steps are then repeated for multiple different POPs.

Brief Summary Text (25):
By populating a database in the foregoing manner, the present invention typically can obtain more current information regarding nodes on the network than is possible with conventional techniques. Such a database can then be used, for example, in connection with identifying geographic locations for clients accessing a website.

Brief Summary Text (26):
In a further aspect, the invention is directed to populating a database with geographic locations for network devices. A node is provided on a network and a connection is made into a network service provider (NSP) point of presence (POP) to obtain a connection to the network via the NSP. A message is then transmitted to the node over the network connection obtained from the NSP. The message is received at the node and a source network identifier is extracted from the message. The route over the network between the node and the POP is then probed to obtain network identifiers for routers along the route. The foregoing steps are then repeated for multiple different POPs.

Brief Summary Text (27):
By generating a database in the foregoing manner, the present invention often can identify important information regarding the topology of the probed network in a more efficient manner than is possible with conventional techniques. Once network identifiers for routers are identified in this manner, such network identifiers can be looked up in a database in an attempt to identify geographic locations for such routers, thereby providing a geographic map of nodes on the network. In addition, using information obtained from probing various routes along the network, information concerning routing patterns on the network often can be derived. Such information may be even further enhanced by providing multiple nodes on the network, sending similar messages to each of such nodes, and then probing the route from the current POP to each of such nodes.

Drawing Description Text (2):
FIG. 1 is a diagram illustrating a communication route between a client and a host server via the Internet.

Drawing Description Text (3):
FIG. 2 is a flow chart illustrating steps for forecasting a geographic position of a network device according to a representative embodiment of the present invention.

Detailed Description Text (2):
The following discussion describes the preferred embodiments of the invention. Wherever possible, the same reference numbers are used throughout the drawings to refer to the same or like parts. The patent applications for "Determining the Geographic Location of a Network Device" and "Network Probing Using Overlapping Packets" filed by Steven Augart concurrently herewith are hereby incorporated by reference as though set forth herein in full.

Detailed Description Text (3):
Geographic Positioning.

Detailed Description Text (4):
An overview of geographic positioning according to a representative embodiment of the invention will be described with reference to the flow diagram illustrated in FIG. 2. Briefly, according to FIG. 2, a request packet is received; if a geographic location has already been determined for the requester, that geographic location is used as the location of the requester; otherwise, the number of hops taken by the request packet is estimated, probe packets are constructed and sent, and responses to the probe packets are received; a check is made for asymmetric routing; the source addresses of received packets are matched to a database to identify the geographic locations of the routers along the path; and the geographic location of the requesting node is identified based on that information.

Detailed Description Text (5):
In more detail, in step 142 a host computer receives a request over a network, such as the Internet, requesting data from the host. The request might be requesting display of a Web page, such as the home page of a particular Web site hosted by the host. However, the first packet of the request typically will be a SYN packet in conformance with the TCP/IP protocols and therefore will implicitly request initiation of a TCP connection between the requesting node and the hosting node.

Detailed Description Text (6):
Immediately upon receipt of such a request (preferably, immediately upon receipt of the SYN packet), the host initiates two parallel paths 143 and 160. In path 143, a geographic location is identified for the requesting node. In path 160, communications continue 162 with the requesting node, typically by first establishing and then utilizing a TCP/IP connection. Ultimately, the geographic location positioning information is supplied from path 143 to step 162 so as to

permit step 162 to generate and transmit to the requester geographic specific information (e.g., local advertising, weather or news). It should be noted that although in the preferred embodiment of the invention, paths 143 and 160 occur simultaneously, it is also possible to defer initiation of path 160 (or at least the first communication of substantive response information in path 160) until the geographic information has been supplied by path 143, so as to permit the very first communication from the host to the requestor to include geographic-specific information.

Detailed Description Text (7):
In step 144, it is determined whether the requestor is already known. In this regard, the request packet preferably includes a source address in accordance with the TCP/IP protocols. By comparing this source address to a database of previously located addresses (e.g., IP addresses), the host can determine whether a geographic position already exists for the current source address (and, in certain embodiments, whether or not that location information has been recently enough updated or is otherwise sufficiently reliable to be presumed valid). If it is determined in step 144 that the requestor is known (and, in certain embodiments, known with adequate accuracy and/or precision), then in step 146 the geographic location of the requestor is retrieved from the database and supplied to the TCP/IP communication process step 162 for use in downloading geographic-specific information to the requester. However, if it is instead determined in step 144 that the requester is unknown (or insufficiently known), then processing proceeds to step 148.

Detailed Description Text (8):
It is noted that the processing performed in step 144 may be as simple as referring to a database to determine whether a geographic position has already been identified. Alternatively, more complicated processing may instead be performed. As explained in more detail below, it is possible that the database may contain multiple geographic positions for the present requestor. If this is the case, a default one of those geographic locations may be selected, one of the geographic locations may be selected based upon predetermined criteria, or the host may attempt to identify the current geographic location, such as by performing the probing technique described below (i.e., commencing with step 148).

Detailed Description Text (9):
Still further, step 144 might identify a geographic home address for the present requestor in the database, but the database might also indicate that the requestor frequently travels. In this case, step 144 might both initiate step 146 to supply the geographic home address of the requestor to TCP/IP communications process 160 and also initiate step 148 to begin the probe packet technique for identifying the current geographic location of the requestor. Communications process 160 may then use either or both of the home address and the current address in its communications with the requester. Finally, step 144 might be omitted entirely and a probing technique (such as the probe packet technique described below) used for each new request.

Detailed Description Text (10):
In step 148, the host computer evaluates the TTL field of the request to estimate how many hops the request has taken to arrive at the host. Preferably, this is accomplished by subtracting the TTL value of the received packet from an assumed initial TTL value for the packet. The TTL field has a maximum value of 255, and most applications set the TTL field to this maximum value so that a packet will make 255 hops before a Time Exceeded packet is returned. Thus, in many cases it can be safely assumed that the number of hops made by the received request packet is equal to 255 minus the value of the TTL field of the received request. On the other hand, it is becoming more common to deviate from the practice of setting the initial TTL value to the maximum of 255. Thus, for example, the current recommendation is to set the value to 60 for outgoing TCP/IP packets. In addition,

certain historical implementations have used initial TTL values of 15 or 30. In order to cope with this variation, the preferred embodiment of the invention utilizes the following technique. If a packet arrives at its destination with a TTL value greater than 195, then it is assumed that the packet's initial TTL was 255. If a packet arrives at its destination with a TTL value of 60 or less, then it is assumed that the initial TTL value was 60. In any other case, or if application of the probing technique with this heuristically determined initial TTL value fails to yield a result, then the system preferably reverts to an alternate method, such as a brute-force method similar to that of conventional traceroute.

Detailed Description Text (12):
In step 150, the host computer constructs and then sends multiple probe packets, each addressed to the source address identified in the request packet. Such probe packets preferably are sent using the User Datagram Protocol (UDP) which is designed for connectionless messages, such as probe packets. It is noted that this is the same protocol used to send probe packets in conventional traceroute operations. Also, as in conventional traceroute techniques, the probe packets preferably are addressed to a port that is unlikely to be in use by the requester, causing a Port Unreachable message to be returned when a probe packet reaches the requester. In fact, subject to the discussion below, it can be assumed that the probe packets sent according to the present invention may have any of the characteristics of probe packets sent according to conventional traceroute operations.

Detailed Description Text (17):
(b) Most ICMP requests should "never trigger an ICMP response", according to RFC 792. In implementations that follow this specification, ICMP Echo requests will never trigger a TTL Exceeded response. Therefore, the ICMP Echo requests, although appropriate for probing the final node in a connection, might not be useful for probing intermediate nodes along the path. Some of the efficiency of the technique for geographic locating according to the present invention comes from its use of a single type of probe packet to elicit responses from both intermediate nodes and the path's destination.

Detailed Description Text (20):
The actual order of sending probe packets (i.e., the bracketing strategy) may change with time as the Internet develops and as prevailing TCP/IP implementations and network policies change. Also, the bracketing strategy will vary if it is determined that the present requestor is likely to be connected to the network via a previously discovered asymmetric routing path or if it is expected that a link in the path commonly uses, or may in the current case be using, multi-path routing. In such a case, the host preferably either performs a full traceroute-like operation or else merely adjusts the value of t used above to account for expected differences in the applicable asymmetric or multi-path routing.

Detailed Description Text (21):
The present invention preferably sends out such a sequence of probe packets without waiting for replies in response to the earlier packets. It is noted that this strategy is different from that of conventional traceroute operations, and can provide for faster identification of the geographic location of the requester.

Detailed Description Text (22):
Finally, the present invention can guard against packet loss by sending out additional probe packets for each particular or for all TTLs in the bracketing strategy. The number of probe packets that are sent at each TTL value will depend upon various engineering trade-offs, such as the value of the location information, the expected datagram loss rates, the cost of bandwidth, the availability of bandwidth, network congestion control policies, and/or whether the initial portion of the route from the server to the client is already known, is of interest, or is even likely to be relevant. Such repeat probe packets can be sent, for example, in

clustered format (e.g., 3 packets with a TTL of t-1, then 3 packets with a TTL of t-2, etc.), in a repeating sequence format (e.g., sending one packet for each TTL value in the sequence and then repeating the entire sequence), or in any other format. Preferably, however, the format is designed to obtain earlier responses from the initial routers in the inbound route and to reduce the statistical likelihood that two or more packets with the same TTL values will be lost.

Detailed Description Text (24):
In step 153, the host checks to determine whether the responses from the initial set of probe packets indicates asymmetric routing, multi-path routing or any other routing anomaly. Such a situation will occur, for example, if the response corresponding to the probe packet having a TTL value of t did not originate from the requestor or if any response corresponding to a TTL value less than t did originate from the requestor. In an ordinary situation with symmetric routing, the response to a probe packet having a TTL value of t (but not the response corresponding to a TTL value of t-1) will be an ICMP Port Unreachable packet. If this is not the case, then it may be determined that either: (i) asymmetric routing, multi-path routing or another routing anomaly has been discovered or (ii) the assumption in step 148 regarding the initial TTL value for the incoming request was wrong. Thus, additional processing may be performed to verify the existence and identify of such anomalous routing. In the event that such a situation is identified, an alternative probing strategy (such as a full "shotgun" approach that runs a modified traceroute program that does not wait between packet transmissions) is performed in step 154 (e.g., using probe packets having TTL values ranging from 1 to 30 or from 1 until an ICMP Port Unreachable packet has been received). Upon completion of such alternative probing operation and verification of an anomalous routing situation or non-standard initial TTL value, the existence of such situation can be stored in a database for future reference (e.g., in step 148 and/or step 150). It is noted that such additional probing for anomaly detection may occur either in real time, offline or using a combination of the two.

Detailed Description Text (25):
In step 155, the source addresses of the responses to the probe packets are matched to a database in an attempt to identify geographic locations of the corresponding routers. Specifically, the server consults a previously collected database of exchange points and finds the last known exchange point closest to the requester. If the last known exchange point can be identified as a launch point for local distribution, then the requesting host can be identified as being in the geographical location serviced by that exchange point. Alternatively, such information can be compared to other geographic information regarding the requestor (e.g., user-supplied information) for confirmation.

Detailed Description Text (26):
Generally, the last known exchange point can be identified by: (i) identifying all responses from the original requestor (e.g., the ICMP Port Unreachable message), (ii) identifying the smallest initial TTL value (N) for a probe packet corresponding to such a response, and then (iii) identifying the response packet corresponding to the probe packet having a TTL value which is one less than N. If the geographic location of the source for this response packet cannot be determined by looking up the source's IP address (or other network identifier) in a database (e.g., because a database entry does not exist for the address or because the database indicates that the associated geographic location may not be sufficiently reliable), then the response packet corresponding to the probe packet having a TTL value of two less than N is examined. This process continues, working back from the original requestor until the geographic location of a router can be reliably determined (or a combination of information for plural routers provides sufficient reliability).

Detailed Description Text (27):
In step 156, the geographic location of the requestor is identified. Generally,

this will be the geographic location of the last known exchange point closest to the requestor. However, as indicated above, in certain embodiments a composite geographic location may be provided based on information for plural different routers. Depending upon how large an area is served by the identified exchange point, the geographic location of the requestor may be known with more or less precision.

Detailed Description Text (28):
For example, if the last known exchange point is within p hops of the requester, the geographic information can be treated as specific information. Otherwise, the geographic information can be treated as a regional indication of where the requestor is located. In this regard, p can be constant, such as 1, 2 or 3. Alternatively, p can be varied based on the indicated geographic region. For example, p might be specified to be a larger number in the Silicon Valley where there are many exchange points and a smaller number in rural areas where there are fewer exchange points within the same geographic region.

Detailed Description Text (29):
In the preferred embodiment of the invention, an improvement over the "Closest Known Neighbor Heuristic" is applied. This heuristic addresses the problem of locating a node x given that x has an unknown location and that one has a map or trace of a packet's path through several nodes including x. If one or more other nodes on the path have known locations, then the heuristic assumes that x is at the same place that the closest (or previous or following) node with a known location is located.

Detailed Description Text (30):
If all of the backbone carriers' nodes at the major exchange points have been located, then it is guaranteed that all unidentified nodes past the last known node must be in the geographic area served by that last known node. Moreover, if the host has a catalogue locating all of such well-known exchange points, or network access points (NAPs), and currently there are few enough of such NAPs that it is practical to catalogue them, then the destination must be served by that NAP. This heuristic is referred to herein as the "Closest Significant Exchange Point Heuristic".

Detailed Description Text (31):
The same heuristic can be applied in a similar manner further down the tree by narrowing the heuristic's geographic scope. For example, if one knows the locations of all of the routers for a cable company's network, then all nodes on the network can be located at the granularity of the individual cable trunk served by each router.

Detailed Description Text (32):
It is also noted that this geographic information can be updated or identified more accurately as additional information is received from the requester. For example, in the event that the requestor submits an on-line purchase order with a delivery address, provides a billing address, or otherwise provides address information, such provided information can be used to more accurately pinpoint the requestor's location. In the event that such provided address information conflicts with the geographic information identified using the probing technique described above, the host may continue to use the geographic information identified by probing or else use the provided address information, as appropriate to the particular circumstance. For instance, if the probe packet technique described above indicates that the requestor is located in Western New York, but the requestor provides a billing address of Raleigh, N.C., the host may continue to use the Western New York information while storing the Raleigh, N.C. address as an alternate address or a possible permanent home address. In this case, after collecting information for a period of time, it may turn out that the requestor's permanent home address does in fact appear to be Raleigh, N.C., but that the requestor frequently travels to

different cities. In this event, depending upon the particular information to be delivered, it may be preferable to utilize both the current temporary address (e.g., for restaurant or store advertisements or for weather information) and the requestor's permanent address (e.g., for advertisements for real estate brokers or for news reports). A conflict might instead mean that the requestor simply had dialed into a POP from a geographic location not ordinarily served by the POP, such as where the requester is traveling or has recently moved but has not updated his POP access telephone number. In any event, the geographic location identified in step 156 is first cached in step 158 and then provided to the communication process of step 160 for use in providing geographic-specific information to the requestor.

Detailed Description Text (33):
In step 158, the identified geographic information is cached for future use. As part of the dynamic database upkeep, data previously acquired about specific IP addresses (including the source IP addresses) are cached for use in the common case of repeated requests (such as return visits to a Web site) from the same IP address. In this case, the route corresponding to the subject IP address has already been traced and the geographic heuristic already applied. This information can be re-used, thus lessening the load on the network and on the server's database and computational resources. However, it may not always be possible to re-use specific geographic information for a particular IP address. For example, IP address allocations may change over time. Thus, the exact geographic address associated with a particular address at one point in time (e.g., such as provided pursuant to an on-line order) sometimes will not correspond to a much later request from the same IP address, as that IP address probably will have been reassigned to a different user. Accordingly, in the preferred embodiment of the invention, such cached information is determined to be relevant only if it has been updated within a specified period of time. The exact time period will vary based on expected tradeoffs, with shorter time periods requiring more frequent probing, together with the attendant overhead, and longer time periods increasing the likelihood that the cached information will be in error.

Detailed Description Text (34):
In step 160, the provided geographic information is used to provide geographic-specific content to the requester, such as local weather, advertising, maps, and places of interest, as well as somewhat more generic information, such as content that is based on time zone (e.g., a "good morning" greeting) or climate in the requestor's region. In addition to using the geographic information obtained above for targeting content, such information may also be used for other purposes, such as marketing research and/or other types of surveys and research.

Detailed Description Text (35):
Additional Considerations Regarding Geographic Locating.

Detailed Description Text (36):
Although the steps are illustrated in a particular order in FIG. 2, it should be understood that the precise order of such steps may be varied to achieve different results, and certain steps may even be performed simultaneously in order to reduce the amount of time required to obtain a geographic position for a node on the network. For instance, any or all of steps 152 through 158 may be initiated prior to completion of step 150 and performed simultaneously with step 150 for a period of time.

Detailed Description Text (37):
In one embodiment, for example, each packet received at step 152 preferably will repeatedly cause steps 153 through 158 to be executed as far as applicable, with step 156 succeeding only once the probes have yielded sufficient information with which to draw a conclusion about the geographic location of the requesting node. Once step 156 has been successfully completed, the concurrently executing step 150 is a synchronously terminated. Any unsent probes in the probe sequence preferably

will not be sent, as they are no longer relevant and generally will only serve to unnecessarily consume resources because the process has already arrived at a conclusion in step 156.

Detailed Description Text (38):
Furthermore, in the preferred embodiment of the invention, step 154, "Perform Modified Probing Operation", will similarly modify the probe sequence being performed by step 150. Specifically, unsent packets in the rest of the original probe sequence will not be sent, and instead a new "rest of the probe sequence" will be constructed (i) from information about what packets have already been sent and (ii) from conclusions reached by step 154 about what probes need to be sent in the modified sequence. Preferably, the information in part (i) is considered in conjunction with that in part (ii), rather than simply naively following the sequence that would be generated by considering only part (ii) information, so that the host can avoid initiating probes that merely unintendedly duplicate probe packets that are already in progress. Nevertheless, the probe sequence generated by in step 154 may deliberately contain duplicates. The integration of information from parts (i) and (ii) referenced above is only intended to avoid sending unintended duplicates that would not otherwise be part of the probe sequence that would be suggested by the part (ii) information alone.

Detailed Description Text (39):
Even at and after step 158, when no new probes are initiated, there often will be probes or replies still traversing the route between the requesting node and the server. It typically will be useful to record any responses the host receives to the outstanding probes in progress, as these serve as additional data for the "slow-side"analysis mechanism. On the other hand, if recording these late responses would impair the performance of the web server or other aspects of the entire system using the geographic locating technique according to the present invention, then any late responses may be simply ignored or discarded. In any case, in the preferred embodiment, waiting for late responses is not blocked. In modern higher level programming languages, this is most conveniently implemented by having the response recording mechanism be a separate process or thread of control within the overall mechanism.

Detailed Description Text (40):
In order to avoid any speed loss from recording the late responses for potential future analysis, one may implement a response recording mechanism, e.g., on a separate computer, that eavesdrops on the network traffic to and from the primary host computer on which the above mentioned process is executed. The separate computer then passively eavesdrops on that network traffic and records probes and their results on some local storage medium for future analysis, without generating any additional network traffic that might slow down the processing being done by the primary host computer.

Detailed Description Text (41):
Creating and Updating the Geographic Database

Detailed Description Text (42):
As indicated above, one feature of the present invention is the creation and use of a database containing geographic information for known exchange points. While it is possible to use multiple databases from various sources that are available over the Internet, accessing such different databases typically will consume additional time, which may be a significant detriment where delivery of real-time geographically targeted content is desired. Accordingly, it is preferable to use a single database and to dynamically update that database as new information is learned.

Detailed Description Text (43):
As the host acquires additional information about requestors' actual locations, for

example by taking on-line orders and noting delivery and billing addresses, these data can be stored to more accurately identify the location of a particular requester. In addition, such data may be compiled and analyzed for common pathways to (perhaps formerly unknown) exchange points, leading to on-the-fly addition of known exchange points. Moreover, if in a given case the result based on responses from the probe packets differs from the location reported by a requester, the database maintenance system may be able to heuristically determine a change in the network's topology. On the other hand, as noted above, if the two locations are sufficiently different, then it may be determined that the requestor utilizes multiple locations (e.g., a permanent home location and a temporary travel location).

Detailed Description Text (44):
Also, in the event that a discrepancy cannot be resolved automatically, the system may notify a human operator of the discrepancy, thus permitting a human database administrator to resolve the discrepancy. As such an administrator begins to identify recurring discrepancies, the resolutions for those discrepancies can be incorporated into the system's heuristics. As a result, over time, more and more of such discrepancies typically will be able to be handled automatically by the system. Still further, the system's heuristics may themselves be updated automatically, e.g., by utilizing known neural network techniques which learn based on the corrections and resolutions made by the human database administrator.

Detailed Description Text (45):
Prior to actual implementation, the database preferably is seeded with available information sources, such as data extracted from the Internic host and other sources of network registrations that are available, lists of locations of exchange points for various ISPs, and reverse telephone lookup information for ISPs' access telephone numbers. Where location/address information is linked in the database, such information preferably is flagged in the database to indicate whether the location associated with each network address is the location where the router for that network address is known to be located or simply one location for an entity that utilizes a number of routers in different geographic locations. ISPs typically fall into the latter category, so that network registration information for an ISP often indicates a single geographic location for all IP addresses assigned to the ISP, even though in actual usage different blocks of such IP addresses may be used by the ISP in different geographic locations. Although such latter type of information may not be capable of immediate direct use, it may be combined with other information (e.g., a list of exchange points for the ISP, reverse telephone lookup information for the ISP's access telephone numbers, and/or network probing information, such as obtained as described above) to yield more precise geographic location information.

Detailed Description Text (51):
4) The client can now send IP packets that the POP will route to the larger Internet. From the perspective of another host on the Internet, the client and POP are just ordinary Internet nodes, with the POP behaving just like any ordinary router between the client and the larger Internet.

Detailed Description Text (52):
Internet Service Providers commonly provide multiple POPs in different geographic locations for their customers. Their intent in doing so is to make it possible for customers to make local telephone calls to the POP, since it is much more attractive for a customer to use the Internet if the customer does not have to pay per-minute toll charges for the telephone connection, in addition to any fees to the ISP. Internet Service Providers commonly distribute lists to their customers of all of their POPs, the telephone access numbers to those POPs, and their geographic locations. These lists are especially useful for customers who travel, because they can often find a local telephone access number regardless of where they are.

Detailed Description Text (53):
The following describes an implementation of a DPL according to a representative embodiment of the invention. Initially, the user of the DPL will arrange accounts with a variety of national and local ISPs. In addition, from these ISPs, the user obtains lists of the modem phone numbers of the ISPs' POPs and the geographic location for each POP. The DPL user may not wish to gather information about all locations; instead, the DPL user may focus on a single region, state, nation, or other area of interest. In this case, the DPL user then selects all of the modem telephone numbers whose locations are in the DPL user's region of interest.

Detailed Description Text (54):
One component of the DPL, the "DPL dialer", then dials (preferably automatically) all, or all selected, numbers on that list. Each time it establishes a PPP link to a POP, the DPL dialer will send an Internet message via that PPP link with a destination address of the DPL data-gathering server (DGS). Actually, the DGS need not be a "server" in a conventional sense, but rather any network device that can receive messages from the dialer. Also, the dialer and the DGS may be housed in the same device, sharing a network connection or having separate connections.

Detailed Description Text (56):
The DPL dialer message generally also will include the IP address temporarily assigned to the client. Accordingly, the DGS associates such source network address with the known geographic location for the POP in its database, preferably by creating a new entry in the database.

Detailed Description Text (58):
In connection with such probing, the DGS records the trace of nodes on the route, as well as the associated geographic location, telephone access phone number, and ISP, and any other debugging or bookkeeping information that may be appropriate to the DPL user's specific needs, to the extent such information has been transmitted to the DGS and is not already stored in a database accessible to the DGS.

Detailed Description Text (63):
In order to gather additional information, in an alternative embodiment of the invention, the DPL is deployed with more than one DGS, in more than one geographic area. The DPL dialer then sends Internet messages to all of the DGSs, and each DGS gathers its own route traces, as described above for a single DGS. These additional route traces provide additional corroborating evidence, permitting confirmation that nodes are in fact determining nodes, rather than nodes that only appear to be determining nodes from one particular direction.

Detailed Description Text (65):
The foregoing technique can provide a database of exchange points at the layer of network topology necessary in order to provide the desired precision for a geographic location. As noted above, a map of NAPs ordinarily suffices to identify the user to the precision of the area served by that NAP, and similar approaches can be used to obtain finer levels of granularity. As used herein, "precision" refers to the granularity at which location information is available, and "accuracy" describes the likelihood that a technique will yield the proper location at a given precision.

Detailed Description Text (67):
Additional methods also can be used for seeding of the location database. Such techniques can be used either individually or in various combinations. For example, the host names of hosts along a traceroute path can be parsed according to the naming conventions used by various ISPs. For example, the following lists certain domain names and their corresponding geographic locations.

Detailed Description Text (80):
Another technique for seeding the location database is to essentially eavesdrop on

the routing tables used by the NAPs. This can be accomplished by consulting the Routing Arbiter Database and/or by eavesdropping on the internal routing protocol (usually RIP or RIP2) traffic at each of the NAPs. For example, by installing a network host at each of the NAPs of interest (such as a small number of the major ones or all of such NAPs) a host according to the present invention can eavesdrop on such traffic. Such special network hosts will then contribute updates to the routing information for use in the host's location database.

Detailed Description Text (81):
With regard to the foregoing, as a general rule it can be asserted that any traffic whose next hop out of the NAP is not to a backbone router, must be to a local network. Therefore, that network's routing prefix (appropriately masked block of network numbers) typically hangs entirely off of that NAP, and is therefore in the metropolitan area served by that NAP. Based on this information, it generally will be a relatively straightforward matter to determine when an IP address is located within an area served by an identified NAP. This technique also may be used to identify a geographic location for a source network address included in an incoming request, i.e., by utilizing the location of a NAP that corresponds to such network address.

Detailed Description Text (83):
In a preferred embodiment, the present invention implements a "hasty listening" technique that allows a TCP application according to the present invention to receive notification of an incoming request sooner than conventional techniques ordinarily would permit. Specifically, in conventional techniques, the TCP application is not notified regarding an incoming communication until after completion of a process known as the "Three-Way Handshake", which is used in establishing a TCP connection. However, according to the hasty listening technique of the present invention, the TCP application is notified as soon as the SYN packet is received, which is the first step in initiating a TCP connection. Hasty listening thus reduces the time during which an application must wait for geographic information about the request's originator to be gathered. Using hasty listening, a TCP application will identify the IP address of the originator of an inbound TCP connection without having to wait for completion of the Three-Way Handshake. As a result, the process of obtaining a geographic location for such requestor (e.g., steps 143, et seq. in FIG. 2) can often be initiated much sooner than conventional techniques would permit, thereby resulting in faster geographic positioning.

Detailed Description Text (84):
It is noted that one method of launching Denial of Service (DoS) attacks is to send TCP SYN packets with a forged source address. The TCP three-way handshake ordinarily will not be completed in the case of such an attack, because the host at the forged source address will know that it did not initiate the handshake to which the TCP listener is responding. However, this type of DoS attack will cause a host using Hasty Listening according to the present invention to waste resources determining the geographic location of a nonexistent peer. Therefore, hosts using Hasty Listening may be more vulnerable to this type of DoS attack.

Detailed Description Text (85):
There are security features in some TCP implementations which detect "SYN Flooding" DoS attacks. These security features will be useful in such a situation. A host using Hasty Listening according to the present invention preferably uses this security feature to aggressively attempt to detect possible SYN Flooding and to stop using Hasty Listening for a time when such a detection has been made.

Detailed Description Text (86):
It is also noted that location information caching, as described above, may reduce the amount of network traffic that a system according to the present invention generates when it receives a DoS attack. This, in turn, often will tend to reduce

the severity of the cascade effect which otherwise would result from repeatedly probing some target that is an innocent additional victim of the DoS attack.

Detailed Description Text (88):
In order to implement the functionality described above, in a preferred embodiment of the present invention, several new application programming interface (API) extensions are defined. In the following discussion, it is assumed that these extensions are being made to the standard "Berkeley Sockets" TCP/IP Application Programming Interface (Sockets API) used, for example, by Unix.TM., Linux.TM., Microsoft.TM. Windows.TM., FreeBSD.TM., and other operating systems. However, it should be understood that the following discussion is not limited to extensions to the Sockets API. Rather, the Sockets API is simply used as an example because currently it is the most common implementation. The names used below also are merely exemplary. It should be noted that the concepts discussed below can be readily extended to other systems and other implementations within those systems, as will be understood by those skilled in the art.

Detailed Description Text (89):
Specifically, in the preferred embodiment of the invention, new extensions are provided in order to effect implementation of hasty listening and to retrieve the TTL value for an incoming request packet. We begin with a discussion of conventional Sockets API. In particular, the following is a typical sequence of Sockets API function calls that a TCP server might make. In the following list, the thread that initiates the function call is indicated in italics and parentheses and is positioned between the function call and the description of the function call. Initially, the server has a single thread of control, the Master Thread.

Detailed Description Text (91):
ioctl (l, . . .) (Master Thread) Sets options for l's behavior and query information about l's status.

Detailed Description Text (97):
Create Worker thread (Master Thread) Usually spawn a worker thread t or subprocess t to handle the data from c. This invokes system functions that are outside the scope of the Sockets API.

Detailed Description Text (99):
select ( ) (Worker t) Check whether there is data to be read on the socket c. The socket c will have data if the remote network peer has sent data to the server over the TCP connection e.

Detailed Description Text (103):
One of the function calls described above, ioctl( ), can be used to query transient information about the connection and set aspects of its state that do not fit neatly into any other standard system operations. The function calling signature for ioctl ( ) is:

Detailed Description Text (105):
The operation parameter can indicate any operation or status query that the API user may wish to perform on the socket. In the case of a status query, one provides an optional-parameter in order to store the results of the query. The interpretation of optional-parameter depends upon the particular operation being performed. In the case of modifying some characteristic of the socket, the use of the optional-parameter depends upon the particular operation, and there are some setting operations that do not require an optional-parameter.

Detailed Description Text (106):
When a user of the Sockets API requests an ioctl( ) operation that is not defined or not appropriate for socket, the API typically returns the standard ENOIOCTLCMD error. Similarly, an API user using the new ioctl( ) operations defined below will

get an ENOIOCTLCMD error if using a TCP implementation that does not support the enhancements described below.

Detailed Description Text (107):
The setsockopt ( ) and getsockopt ( ) API calls are intended to modify more persistent aspects of sockets. Their function calling signatures are similar to that of ioctl ( ) and are not discussed here in detail. There are well-known socket options that are common to all implementations of the Sockets API. In addition, each implementation of the TCP/IP Sockets API may have its own implementation-specific named/defined operations.

Detailed Description Text (108):
When a user of the Sockets API requests a setsockopt ( ) or getsockopt ( ) operation that is not defined, the API returns the ENOPROTOOPT error. Similarly, an API user using the setsockopt ( ) and getsockopt ( ) operations, defined below, will get the ENOPROTOOPT error if using a TCP Implementation that does not support the enhancements described below.

Detailed Description Text (111):
select ( ) checks whether one or more sockets can have certain API calls made on them without that call "blocking". If an API call "blocks", that means that the TCP implementation must wait for something to happen before that API call can be completed.

Detailed Description Text (112):
select ( ) is important for programs that need to interact on several communication channels at the same time. There are many programs that use the Sockets API and never call select ( ); these programs all use a single communication channel at a time. When these programs make potentially blocking calls to Sockets API functions, they typically wait until the calls complete.

Detailed Description Text (113):
The common use of select( ) in the Sockets API is to check whether data is available to be read on one or more connected TCP sockets. If, when select( ) returns, it has indicated that some connected TCP socket, c, is among the readable-sockets, then a program calling the standard Sockets API read( ) function call on c will get back data immediately. The socket c will have data if the TCP implementation has received data over c's TCP connection, e, and has not yet returned that data in a previous use of read( ).

Detailed Description Text (114):
Most Sockets API programmers only use select ( ) on sockets or other I/O descriptors that are connected to a single established TCP connection or other established I/O channel. However, there is a special use of the select( ) readable-sockets argument, which we discuss here.

Detailed Description Text (115):
In classic Sockets API programs, such as \`inetd,\` after listen ( ) has been called on a socket l, select ( ) is often called with the listen( )ing socket, l, as one of the maybe-readable-sockets. When a new connection, e, is established to l's TCP port, then e is added to l's "established connection backlog". If there is an established connection in l's "established connection backlog", such that an accept ( ) operation will immediately return a usable connected socket, c, then l is considered "readable" for the purposes of select ( ).

Detailed Description Text (117):
It is noted that l is not actually "readable" in the normal sense of the word; a read( ) API call on l will fail. However, select( ) uses the readable-sockets parameter to represent information about l's connection backlog. This is an arbitrary convention; the writable-sockets or exceptional-conditions-sockets

parameters would (arguably) have been equally appropriate choices.

Detailed Description Text (118):
As noted above, the extensions to the API according to the present invention allow two new capabilities: hasty listening and TTL value retrieval.

Detailed Description Text (119):
Hasty Listening API Extension.

Detailed Description Text (130):
The lower level details of how Sockets API functions indicate error codes are as follows: Sockets API functions, such as ioctl( ) or setsockopt( ), return the constant integer value (-1); -1, by convention, indicates that some error occurred, but does not specify the error. ioctl( ) and setsockopt( ) specify the particular error by setting a standard data storage area, errno, to a small integer indicating one of the predefined system error codes. In the case of the ENOMSG error, we set the errno area to the constant ENOMSG, which is defined in a system file (include/errno.h).

Detailed Description Text (134):
API Extension for Retrieval of TTL Information.

Detailed Description Text (144):
Generally, the methods described herein will be practiced with a general purpose computer, either with a single processor or multiple processors. FIG. 3 is a block diagram of a general purpose computer system, representing one of many suitable computer platforms for implementing the methods described above. FIG. 3 shows a general purpose computer system 250 in accordance with the present invention. As shown in FIG. 3, computer system 250 includes a central processing unit (CPU) 252, read-only memory (ROM) 254, random access memory (RAM) 256, expansion RAM 258, input/output (I/O) circuitry 260, display assembly 262, input device 264, network interface 265, and expansion bus 266. Computer system 250 may also optionally include a mass storage unit 268 such as a disk drive unit or nonvolatile memory such as flash memory and a real-time clock 270.

Detailed Description Text (146):
I/O circuitry 260 typically includes a number of latches, registers and direct memory access (DMA) controllers. The purpose of I/O circuitry 260 is to provide an interface between CPU 252 and a network via network interface 265 and such peripheral devices as display assembly 262, input device 264, and mass storage 268.

Detailed Description Text (150):
A removable storage read/write device 269 may be coupled to I/O circuitry 260 to read from and to write to a removable storage medium 271. Removable storage medium 271 may represent, for example, a magnetic disk, a magnetic tape, an opto-magnetic disk, an optical disk, or the like. Instructions for implementing the inventive method may be provided, in one embodiment, to a network via such a removable storage media.

Detailed Description Text (152):
Expansion bus 266 is coupled to data bus 272, control bus 274, and address bus 276. Expansion bus 266 provides extra ports to couple devices such as network interface circuits, modems, display switches, microphones, speakers, etc. to CPU 252. Network communication is accomplished through the network interface circuit 265 and an appropriate network.

Detailed Description Text (156):
Although the present invention has been described in reference to an embodiment for use on the Internet, it should be understood that those embodiments are exemplary

only and that other embodiments for use on other networks can also be provided in accordance with the teachings set forth above. In particular, it should be understood that references above to the Internet should be understood to apply equally to other networks, such as other packet switched networks. Similarly, in the more general case, references to IP addresses above can be replaced with references to network addresses or network identifiers, and other similar generalizations can be made, as will be understood by those skilled in the art.

CLAIMS:

1. A method of populating a database with geographic locations for network devices, said method comprising: a. providing a node on a network; b. connecting into a network service provider (NSP) point of presence (POP) to obtain a connection to the network via the NSP; c. transmitting a message to the node over the network connection obtained from the NSP; d. receiving the message at the node and extracting a source network identifier from the message; e. associating the source network identifier with a known geographic location for the POP in a database; and f. repeating steps (b) through (e) for plural different POPs.

2. A method according to claim 1, further comprising a step of: (g) probing a route over the network between the node and the POP to obtain network identifiers for routers along the route.

3. A method according to claim 2, wherein said probing in step (g) comprises transmitting plural probe packets addressed to the network identifier extracted from the message.

4. A method according to claim 2, wherein said probing in step (g) comprises transmitting plural probe packets addressed to said node on the network.

6. A method according to claim 2, further comprising a step of: (h) looking up the network identifiers for said routers in the database in an attempt to identify geographic locations for said routers.

7. A method according to claim 2, further comprising a step of: (h) extracting geographic locations for said routers from responses to said probing performed in step (g).

8. A method according to claim 2, wherein said probing in step (g) is performed by said node on the network.

10. A method according to claim 1, wherein step (d) further comprises a substep of: (d1) in response to the message, probing a route over the network between the POP and the node to obtain network identifiers for routers along the route.

11. A method according to claim 10, wherein said probing in step (d1) comprises transmitting plural probe packets addressed to the network identifier extracted from the message, and wherein said plural probe packets have different initial Time-To-Live (TTL) values.

16. A method according to claim 10, wherein said probing in step (d1) comprises transmitting plural probe packets addressed to the network identifier extracted from the message.

18. A method according to claim 10, wherein step (d) further comprises a substep of: (d2) looking up the network identifiers for said routers in the database in an attempt to identify geographic locations for said routers.

19. A method according to claim 10, further comprising a step of aggregating network nodes into groups.

21. A method according to claim 19, further comprising a step of identifying a determining node on the network, based on plural probing results, wherein when probing a route to a source node, presence of the determining node in an identified route strongly suggests that the source node is in a particular group.

22. A method according to claim 21, wherein the particular group consists of network nodes in a specified geographic area.

23. A method according to claim 8, wherein plural nodes are provided in step (a), and the route over the network from the POP to each of said plural nodes is probed in step (d1).

25. A method according to claim 10, further comprising a step of identifying a determining node on the network, based on plural probing results, wherein when probing a route to a source node, presence of the determining node in an identified route strongly suggests that the source node is in a particular group.

26. A method according to claim 1, wherein the network is an internet.

27. A method of populating a database with geographic locations for network devices, said method comprising: a. providing a node on a network; b. connecting into a network service provider (NSP) point of presence (POP) to obtain a connection to the network via the NSP; c. transmitting a message to the node over the network connection obtained from the NSP; d. receiving the message at the node and extracting a source network identifier from the message; e. probing a route over the network between the node and the POP to obtain network identifiers for routers along the route; and f. repeating steps (b) through (e) for plural different POPs.

28. A method according to claim 27, wherein said probing in step (e) comprises transmitting plural probe packets addressed to the network identifier extracted from the message.

30. A method according to claim 27, further comprising a step of: (g) looking up the network identifiers for said routers in the database in an attempt to identify geographic locations for said routers.

31. A method according to claim 27, further comprising a step of: (g) extracting geographic locations for said routers from responses to said probing performed in step (e).

32. A method according to claim 27, wherein plural nodes are provided in step (a), and the route over the network from the POP to each of said plural nodes is probed in step (e).

33. A method according to claim 27, wherein said probing in step (e) comprises transmitting plural probe packets addressed to said node on the network.

Previous Doc Next Doc Go to Doc#